

Задача А. Компоненты реберной двусвязности

Имя входного файла: `bicone.in`
Имя выходного файла: `bicone.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Компонентой реберной двусвязности графа $\langle V, E \rangle$ называется подмножество вершин $S \subset V$, такое что для любых различных u и v из этого множества существует не менее двух реберно не пересекающихся путей из u в v .

Дан неориентированный граф. Требуется выделить компоненты реберной двусвязности в нем.

Формат входных данных

Первая строка входного файла содержит два натуральных числа n и m — количества вершин и ребер графа соответственно ($1 \leq n \leq 20\,000$, $1 \leq m \leq 200\,000$).

Следующие m строк содержат описание ребер по одному на строке. Ребро номер i описывается двумя натуральными числами b_i, e_i — номерами концов ребра ($1 \leq b_i, e_i \leq n$).

Формат выходных данных

В первой строке выходного файла выведите целое число k — количество компонент реберной двусвязности графа. Во второй строке выведите n натуральных чисел a_1, a_2, \dots, a_n , не превосходящих k , где a_i — номер компоненты реберной двусвязности, которой принадлежит i -я вершина.

Примеры

<code>bicone.in</code>	<code>bicone.out</code>
6 7 1 2 2 3 3 1 1 4 4 5 4 6 5 6	2 1 1 1 2 2 2

Задача В. Магнитные подушки

Имя входного файла: `city.in`
Имя выходного файла: `city.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Город будущего застроен небоскребами, для передвижения между которыми и парковки транспорта многие тройки небоскребов соединены треугольной подушкой из однополярных магнитов. Каждая подушка соединяет ровно 3 небоскреба и вид сверху на нее представляет собой треугольник, с вершинами в небоскребах. Это позволяет беспрепятственно передвигаться между соответствующими небоскребами. Подушки можно делать на разных уровнях, поэтому один небоскреб может быть соединен различными подушками с парами других, причем два небоскреба могут соединять несколько подушек (как с разными третьими небоскребами, так и с одинаковым). Например, возможны две подушки на разных уровнях между небоскребами 1, 2 и 3, и, кроме того, магнитная подушка между 1, 2, 5.

Система магнитных подушек организована так, что с их помощью можно добираться от одного небоскреба, до любого другого в этом городе (с одной подушки на другую можно перемещаться внутри небоскреба), но поддержание каждой из них требует больших затрат энергии.

Требуется написать программу, которая определит, какие из магнитных подушек нельзя удалять из подушечной системы города, так как удаление даже только этой подушки может привести к тому, что найдутся небоскребы из которых теперь нельзя добраться до некоторых других небоскребов, и жителям станет очень грустно.

Формат входных данных

В первой строке входного файла находятся числа N и M — количество небоскребов в городе и количество работающих магнитных подушек соответственно ($3 \leq N \leq 100000$, $1 \leq M \leq 100000$). В каждой из следующих M строк через пробел записаны три числа — номера небоскребов, соединенных подушкой. Небоскребы пронумерованы от 1 до N . Гарантируется, что имеющиеся воздушные подушки позволяют перемещаться от одного небоскреба до любого другого.

Формат выходных данных

Выведите в выходной файл сначала количество тех магнитных подушек, отключение которых невозможно без нарушения сообщения в городе, а потом их номера. Нумерация должна соответствовать тому порядку, в котором подушки перечислены во входном файле. Нумерация начинается с единицы.

Примеры

city.in	city.out
3 1 1 2 3	1 1
3 2 1 2 3 3 2 1	0
5 4 1 2 3 2 4 3 1 2 4 3 5 1	1 4

Задача С. Мастерство тракториста

Имя входного файла:	<code>tractor.in</code>
Имя выходного файла:	<code>tractor.out</code>
Ограничение по времени:	10 секунд
Ограничение по памяти:	512 мегабайт

Вредный культорг Ишма Дровкин обожает срывать учебные процессы. Особенно он любит заставлять школьников проявлять физическую активность. Вот сейчас, проявив недюжинный ораторский талант, он убедил школьников угнать трактор с хутора и отправиться кататься на нём по полю. Однако, трактор, который им предстоит угнать, старше половины преподавательского состава параллели В, вместе взятой, поэтому им невозможно рулить. Он может ехать вперёд и только вперёд.

Более формально — представим поле в форме прямоугольника $H \times W$. Некоторые клетки этого поля заблокированы и по ним невозможно проехать даже на тракторе. Трактор может быть повернут вдоль направления движения по вертикали, горизонтали или диагонали.

Школьники в ЛКШ — люди не только упорные, но и развитые физически, поэтому их не смущает перспектива разворачивать трактор своими руками каждый раз, когда им требуется повернуть. Но, естественно, это не самое приятное времяпрепровождение, так что они просят вас найти маршрут, следуя по которому они будут вынуждены работать руками минимальное количество раз. Важная деталь заключается в том, что сейчас трактор валяется на поле в перевёрнутом состоянии, так что по меньшей мере один раз им придётся повернуть трактор в правильное положение, которое они могут выбрать удобным им образом. То, как именно будет повернут трактор в конечной точке, не имеет значения.

Формат входных данных

В первой строке входного файла заданы целые числа H, W ($1 \leq H, W \leq 1000$) — высота и ширина поля.

В последующих H строках задано поле, заблокированные клетки обозначены латинскими буквами "X", а свободные — точками.

После поля следует строка с двумя целыми числами sx, sy ($1 \leq sx \leq W, 1 \leq sy \leq H$) — координаты стартового положения трактора.

Последней строкой идут два целых числа tx, ty ($1 \leq tx \leq W, 1 \leq ty \leq H$) — координаты конечного положения трактора.

Координаты отсчитываются от нижнего левого угла поля. Стартовое положение не совпадает с конечным. Стартовая и конечная клетки не являются заблокированными.

Формат выходных данных

Если Ишма Дровкин настолько большой шутник, что пути между стартовой и конечной клетками не существует, выведите -1 . Иначе выведите единственное натуральное число — минимальное количество поворотов, которые школьники должны будут сделать по пути к конечной клетке.

Примеры

tractor.in	tractor.out
5 7 XX...X X.XXX.. ..XXX.X X.X...X ...XXX 1 1 6 5	3
4 3 .XX .XX XX. X.. 1 3 3 2	-1

Замечание

В примере школьники сначала перевернут трактор и развернут его направо. Потом они проедут три клетки и повернут трактор на 45 градусов против часовой стрелки, чтобы тот смотрел по диагонали вверх и вправо. Затем они проедут две клетки и снова повернут трактор на 45 градусов против часовой стрелки, чтобы тот смотрел наверх. Наконец, они проедут три клетки вверх и окажутся в конечной клетке.

Задача D. Авиаперелёты

Имя входного файла: `avia.in`
Имя выходного файла: `avia.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Главного конструктора Петю попросили разработать новую модель самолёта для компании «Air Бубундия». Оказалось, что самая сложная часть заключается в подборе оптимального размера топливного бака.

Главный картограф «Air Бубундия» Вася составил подробную карту Бубундии. На этой карте он отметил расход топлива для перелёта между каждой парой городов.

Петя хочет сделать размер бака минимально возможным, для которого самолёт сможет долететь от любого города в любой другой (возможно, с дозаправками в пути).

Формат входных данных

Первая строка входного файла содержит натуральное число n ($1 \leq n \leq 1000$) — число городов в Бубундии. Далее идут n строк по n чисел каждая. j -е число в i -й строке равно расходу топлива при перелёте из i -го города в j -й. Все числа не меньше нуля и меньше 10^9 . Гарантируется, что для любого i в i -й строчке i -е число равно нулю.

Формат выходных данных

Первая строка выходного файла должна содержать одно число — оптимальный размер бака.

Примеры

avia.in	avia.out
4 0 10 12 16 11 0 8 9 10 13 0 22 13 10 17 0	10

Задача Е. Разбиение на пути

Имя входного файла: `paths.in`
Имя выходного файла: `paths.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Разбейте рёбра неориентированного графа на минимальное количество рёберно-простых путей.

Формат входных данных

Дан граф. На первой строке число вершин n ($1 \leq n \leq 20\,000$) и число рёбер m ($1 \leq m \leq 20\,000$).
Следующие m строк содержат описание рёбер графа. Каждая строка по два числа a_i и b_i ($1 \leq a_i, b_i \leq n$). Между каждыми двумя вершинами не более одного ребра. Граф связан.

Формат выходных данных

На первой строке минимальное число путей. На каждой следующей описании очередного пути – номера вершин в порядке прохождения.

Примеры

<code>paths.in</code>	<code>paths.out</code>
7 7 1 2 4 1 6 7 5 7 7 4 2 3 4 2	3 2 4 7 6 7 5 4 1 2 3